## Client-Server Chat Application - Project Report Course: Computer Networks (CS305)

Instructor: Ms. Poma Panezai Group Members: [Your Names Here] Semester: 2nd, Spring 2025

Date Submitted: June 5, 2025

# Abstract

This report presents the design and implementation of a Java-based Client-Server Chat Application for the CS305 course. The system allows multiple clients to connect to a centralized server and exchange messages in real-time. It uses TCP sockets and multithreading to ensure efficient communication and scalability. The GUI is built using Java Swing to enhance usability and visualization. This project demonstrates core networking concepts and provides hands-on experience with socket programming in Java.

# Introduction

Communication systems are a fundamental aspect of modern networking. This project focuses on implementing a basic chat system where clients communicate through a central server. Using Java, we aim to simulate a simplified version of commercial messaging applications with a focus on multi-client handling and real-time data exchange.

# Problem Statement

In real-world networks, message transmission between users requires a centralized infrastructure that can manage multiple clients, handle data synchronization, and ensure message delivery. Our project addresses this by creating a server that can handle concurrent client connections and route messages appropriately.

# Objectives

* Implement a server that can accept multiple client connections using multithreading.
* Create a client GUI that allows message input and displays incoming messages.
* Ensure real-time message broadcasting to all connected clients.
* Use TCP sockets for reliable message transmission.
* Apply basic principles of network programming and GUI design in Java.

# System Design

* 1. Architecture

The system uses a centralized server model. Each client connects to the server via a TCP socket. The server spawns a new thread for each client and uses a shared collection to broadcast messages to all connected clients.

# Implementation

* Programming Language: Java
* IDE: Apache NetBeans
* Technologies Used: Java Sockets, Java Swing, Multithreading
* Files:

**Server.java Client.java**

: Contains server logic and GUI for monitoring.

: Contains client-side logic and GUI for sending/receiving messages.

The server listens on a fixed port (12345). Upon connection, it creates a new thread for each client and maintains a shared list of output streams to broadcast messages.

# Testing & Results

* The application was tested using multiple client instances.
* Clients can send and receive messages instantly.
* The server displays log messages for all connected clients.
* No connection loss was observed during tests.

Screenshots:

[Insert screenshots of the server and client GUI here]

# Conclusion

This project successfully demonstrates the use of Java sockets and multithreading to build a basic real-time chat application. The GUI ensures ease of use while maintaining network functionality. This implementation helped solidify our understanding of client-server communication, thread management, and user interface design.

# References

* GeeksforGeeks - Java Socket Programming
* Oracle Java Documentation
* Stack Overflow Discussions
* ChatGPT for code generation and debugging help

# GitHub Repository

https://github.com/Joshwaop/ChatApplication